CS41 Homework 6

This homework is due at 11:59PM on Sunday, November 3. Write your solution using LaTeX. Submit this homework in a file named `hw6.tex` using **github**. This is a **10 point assignment**. This is a partnered homework. You should primarily be discussing this homework with your partner.

It's ok to discuss approaches at a high level. In fact, I encourage you to discuss general strategies. However, you should not reveal specific details of a solution, nor should you show your written solution to anyone else. The only exception to this rule is work you've done with a lab partner/group *while in lab*. In this case, note (in your post-homework survey) who you've worked with and what parts were solved during lab.

The main **learning goals** of this homework are to get more practice with algorithm design using divide and conquer and dynamic programming.

1. **Divide and conquer minimum spanning trees?**

   Lila has a really cool idea for a divide and conquer algorithm which will find a MST. Given a connected, undirected graph $G = (V, E)$ with weighted edges, Lila's algorithm does the following:

   - Divides the graph into two pieces, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. ($V_1 \cup V_2 = V$ and V_1 and V_2 are disjoint. E_1 is the edges in E with both endpoints in V_1, and E_2 is the edges in E with both endpoints in V_2.)
   - Recursively finds the MSTs M_1 for G_1 and M_2 for G_2.
   - Finds the lowest-weight edge $e = (u, v)$ with $u \in V_1$ and $v \in V_2$.
   - Returns the minimum spanning tree $M_1 \cup M_2 \cup \{e\}$.

   Unfortunately, this algorithm does not work. Give an example input graph G with weights and describe a run of this algorithm where the algorithm does not return a minimum spanning tree on G.

2. **Integer Multiplication.** Recall the Karatsuba algorithm for integer multiplication from class, which multiplies two n-digit base-N numbers a, b by:

$$\begin{aligned} a \cdot b &= (a_L N^{n/2} + a_r)(b_L N^{n/2} + b_R) \\ &= a_L b_L N^n + (a_L b_R + a_R b_L) N^{n/2} + a_R b_R \\ &= AN^n + (C - A - B) N^{n/2} + B, \end{aligned}$$

   where the three $(n/2)$-digit multiplications are:

   - $A = a_L \cdot b_L$
   - $B = a_R \cdot b_R$
   - $C = (a_L + a_R)(b_L + b_R)$

   Consider an algorithm for integer multiplication of two n-digit base-N numbers where each number is split into three parts, each with $n/3$ digits.

(a) Design such an algorithm, similar to the integer multiplication we did in class. Your algorithm should describe how to multiply two integers using only six multiplications (instead of the straightforward nine).

(b) Determine an asymptotic upper bound for the running time of your algorithm. (Write it as a recurrence, and then solve the recurrence.) You should use partial substitution.

(c) Is this algorithm asymptotically faster than Karatsuba multiplication? That is, is it better to use the algorithm that breaks an integer into three parts, or two parts?

(d) **(extra challenge)** Suppose there were an algorithm that used only five multiplications of $n/3$-digit numbers instead of six. Determine an asymptotic upper bound for the running time of such an algorithm. Would this algorithm be asymptotically faster than Karatsuba multiplication?

3. **Cassie's Convenience Stores.** Carrie plans to open a chain of convenience stores along Baltimore Pike. Using market research, Carrie identified a series of n locations where she can open stores. For each location, Cassie calculated (again using market research) how much annual profit she is likely to gain by placing a store at this location. She can build as many convenience stores as she wants, as long as they are not too close (otherwise, they will compete with each other for business and lose money).

   In this problem, you will design an algorithm that helps Cassie determine how much annual profit she can make. The input to this problem is an integer K, and two arrays $L[1 \cdots n]$ and $P[1 \cdots n]$. Assume that $0 \leq L[i] \leq N$ for each i[1], and that L is sorted in increasing order. The goal of this problem is to output the maximum possilbe profit by placing convencience stores at locations from $L[1 \cdots n]$ such that the distance between any two locations is at least K.

   (a) If Cassie decides to build a convenience store at location $L[k]$, what is the closest location to the east or west that she can build, given her list of locations? Write an algorithm $West[k]$ that returns the index of the closest location k' to k such that $L[k'] < L[k]$. Write a similar algorithm for $East[k]$.

   (b) Design a dynamic program that computes the maximum annual profit Cassie can earn by placing her convenience stores.

   (c) Modify your dynamic program so it returns the set of locations that maximize Cassie's profit.

[1] $L[i]$ represents the the location of store i, where 0 is the westernmost terminus of Baltimore Pike, and N is the easternmost terminus